# Notes and Erratum

## Where to Instantiate Models

In the book, we instantiate models at several different points in the code. This approach can make it tricky to understand which instantiation option is best.

The best practice, based on options available to SAPUI5 developers today, is to have most models automatically instantiated by defining them in the *manifest.json*.

However, there is one limitation to the approach: Only the models that are required and used throughout the entire app should go into the manifest.

Other models, like the JSON models, are known as *view models* within the book. These are usually only used in a single view, and should hence be instantiated from the view's respective controllers.

The special *device model* touched upon in Chapter 5, Section 5.9.2 is better instantiated in the component, as you do not rely on a particular data source to serve as model data, but rather use the device API shipped with SAPUI5. This needs to be loaded as a dependency in the component. It makes sense to only create this model when you can make sure the necessary information from this API is indeed already loaded.

## Chapter 5: Missing Supplier Products

The *suppliers.json* file is supposed to contain products in this chapter. The book is missing the part where they should have been added. To work through this chapter, please make sure your *suppliers.json* file now looks like below:

```
{
  "Suppliers":[
    {
      "id":0,
      "Name":"Exotic Liquids",
      "Address":{
        "Street":"NE 228th",
        "City":"Sammamish",
        "State":"WA",
        "ZipCode":"98074",
        "Country":"USA",
        "PhoneNumber": "+1-123-123-1234"
      },
      "Location":{
        "type":"Point",
        "coordinates":[
          -122.03547668457,
          47.6316604614258
        ]
      },
      "Products":[
        {
          "id":1,
          "Name":"Milk",
          "Description":"Low fat milk",
          "ReleaseDate":"1995-10-01T00:00:00",
          "DiscontinuedDate":null,
          "Allergenics": "",
          "Rating":3,
          "Price":3.5
        },
        {
          "id":2,
          "Name":"Vint soda",
          "Description":"Americana Variety - Mix of 6
flavors",
          "ReleaseDate":"2000-10-01T00:00:00",
          "DiscontinuedDate":null,
          "Allergenics": "",
          "Rating":3,
          "Price":20.9
        },
        {
          "id":3,
          "Name":"Havina Cola",
          "Description":"The Original Key Lime Cola",
          "ReleaseDate":"2005-10-01T00:00:00",
          "DiscontinuedDate":"2006-10-01T00:00:00",
          "Allergenics": "",
          "Rating":3,
          "Price":19.9
        },
        {
          "id":4,
          "Name":"Fruit Punch",
          "Description":"Mango flavor, 8.3 Ounce Cans (Pack
of 24)",
          "ReleaseDate":"2003-01-05T00:00:00",
          "DiscontinuedDate":null,
          "Allergenics": "",
          "Rating":3,
          "Price":22.99
        },
        {
          "id":5,
          "Name":"Cranberry Juice",
          "Description":"16-Ounce Plastic Bottles (Pack of
12)",
          "ReleaseDate":"2006-08-04T00:00:00",
          "DiscontinuedDate":null,
          "Allergenics": "",
          "Rating":3,
          "Price":22.8
        },
        {
          "id":6,
          "Name":"Pink Lemonade",
          "Description":"36 Ounce Cans (Pack of 3)",
          "ReleaseDate":"2006-11-05T00:00:00",
      "DiscontinuedDate":null,
      "Allergenics": "Milk, Soy, Edible Nuts, Gluten",
      "Rating":3,
      "Price":18.8
    }
  ]
},
{
  "id":1,
  "Name":"Tokyo Traders",
  "Address":{
     "Street":"NE 40th",
     "City":"Redmond",
     "State":"WA",
     "ZipCode":"98052",
     "Country":"USA",
     "PhoneNumber": "+1-123-123-1235"
   },
   "Location":{
     "type":"Point",
     "coordinates":[
       -122.107711791992,
       47.6472206115723
     ]
   },
   "Products":[
     {
       "id":7,
       "Name":"Bread",
       "Description":"Whole grain bread",
       "ReleaseDate":"1992-01-01T00:00:00",
       "DiscontinuedDate":null,
       "Allergenics": "",
       "Rating":4,
           "Price":2.5
         },
         {
           "id":8,
           "Name":"DVD Player",
           "Description":"1080P Upconversion DVD Player",
           "ReleaseDate":"2006-11-15T00:00:00",
           "DiscontinuedDate":null,
           "Allergenics": "Milk, Soy, Edible Nuts, Gluten",
           "Rating":5,
           "Price":35.88
         },
         {
           "id":9,
           "Name":"LCD HDTV",
           "Description":"42 inch 1080p LCD with Built-in
Blu-ray Disc Player",
           "ReleaseDate":"2008-05-08T00:00:00",
           "DiscontinuedDate":null,
           "Allergenics": "Milk, Soy, Edible Nuts, Gluten",
           "Rating":3,
           "Price":1088.8
         }
       ]
     }
   ]
 }
```

*Listing 1: suppliers.json with Supplier Products Included*

## Chapter 5: The Base Controller Concept

In Chapter 5, controllers within the sample application are using a base controller the other controllers now inherit from, instead of directly inheriting from `sap.ui.core.mvc.Controller`.

The concept behind this is simply to move functionality which is used across controllers within an application to a single file, so other controllers can inherit from it and there is less code redundancy. The base controller is just a normal controller in SAPUI5, and contains convenience functions like `getRouter`, `getModel` etc., which wrap functionality from SAPUI5 that would otherwise be more verbose when used. As an example, the function `getResourceBundle` wraps this code:

```
this.getOwnerComponent().getModel("i18n").getResourceBundle(
);
```

Whenever you have functionality you see becoming redundant in your controller, it's time to create a parent controller. You can of course also have several parent controllers in your application. This means, when your app consists basically of two different chunks of functionality, so that five of your controllers share a certain number of functions and the other six controllers share a different implementation, you may want to create two different parent controllers to inherit from.